



## Advanced Numerical Simulations of Lid-Driven Cavity Flows Using an Optimized Parallelized Fractional Step Method

Muhammad Abid<sup>1,\*</sup>, Tayyaba Akhtar<sup>2</sup>, Mohamed M. Awad<sup>3</sup> and Muhammad Imran<sup>4,5</sup>

<sup>1</sup> Department of Mathematics, North Carolina State University, Raleigh, 27695 NC, United States; mabid@ncsu.edu,;

<sup>2</sup> Department of Mathematics, Government College University Faisalabad, Faisalabad 38000, Pakistan; tayyabakanwal10@gmail.com,;

<sup>3</sup> Mechanical Power Engineering Department, Faculty of Engineering, Mansoura University, Mansoura, Egypt; m\_m\_awad@mans.edu.eg,;

<sup>4</sup> Department of Mathematics, Government College University Faisalabad, Faisalabad 38000, Pakistan; drmmimranchaudhry@gcuf.edu.pk,;

<sup>5</sup> Biruni University, Education Faculty, Department of Mathematics and Science Education, İstanbul Turkey.

### Citation:

Received: 02 September 2024  
Revised: 21 November 2024  
Accepted: 06 December 2024

Abid, M., khtar, T., Awad M. M., and Imran, M. (2025). Advanced Numerical Simulations of Lid-Driven Cavity Flows Using an Optimized Parallelized Fractional Step Method. *Computational Algorithms and Numerical Dimensions*, 4 (1), 1-17.

### Abstract

This work presents a numerical simulation of lid-driven cavity flows by solving the incompressible Navier-Stokes equations, with results validated against established benchmarks. The developed code uses a fractional step method combined with second-order Adams-Bashforth time discretization and Crank-Nicolson spatial discretization to achieve both accuracy and stability. A finite difference method is applied on a staggered grid to implement key operators such as gradient, divergence, Laplace, and advection, all computed with second-order central differencing. To efficiently solve the pressure Poisson equation, the conjugate gradient algorithm is employed. The code is vectorized and parallelized to enhance computational performance. An intriguing case involving opposite moving walls is also explored, revealing altered flow dynamics. This robust and efficient CFD code enables fast, accurate simulations of various fluid flow cases, providing deeper insights into complex flow phenomena.

**Keywords:** Lid-driven cavity flow, Finite difference method, Fractional step method, Numerical simulations, Parallel computing.



Corresponding Author: mabid@ncsu.edu



<https://doi.org/10.22105/cand.2024.488797.1159>



Licensee System Analytics. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0>).

# 1|Introduction

Specifically simulating incompressible fluid flows in the such type of the confined geometries that is critical for both foundational fluid mechanics research and as well as also for the myriad engineering applications. The canonical lid-driven cavity framework serves as a one of the significant benchmark for validating computational fluid dynamics (CFD) capabilities. Nevertheless, conceptually that is very simple, cavity flows show surprisingly complex behaviors like as recirculating eddies and as well as the shear layer instabilities at higher Reynolds numbers. Mathematically speaking, such kind of the flows that are governed by the Navier-Stokes equations which constitute significant numerical challenges due to their non-linear nature in the coupled behavior. So proficiently solving them requires robust kind of the algorithms and as well as discretization schemes that balance accuracy of the numerical method, with the stability and also convergence speed. This demands an in-depth understanding of finite difference formulations, solution procedures, and with the very important factors impacting on the computational cost [1]. The lid-driven cavity flow problem has been comprehensively studied both of experimentally and on the other side as a numerically over the past many decades. It requires fluid motion in a square cavity induced by the translation of the top side of the boundary.

In the face of its geometric simplicity, this flow demonstrate a very rich physics including the primary, secondary, and also tertiary vortices significantly depending on the Reynolds number. At low Reynolds numbers, a primary vortex immerses nearly all of the cavity. As the Reynolds's number increases within the process, secondary vortices appear in the bottom corners, also growing in size and as well as strength. Further increases lead to additional vortices and ultimately transition to turbulence. The benchmark numerical results by Ghia et al. [2] have been presented as a key reference point for validating new computational methods for this canonical problem. The Navier-Stokes equations governing incompressible cavity flows serve unique numerical challenges. The non-linear advection terms introduce instabilities that must be rigorously handled. The incompressibility constraint leads to a pressure-velocity coupling that requires special treatment. Numerous numerical schemes have been developed to address these kind of the issues, including the fractional step method, projection methods, and also coupled solvers [3].

The fractional step approach, pioneered by Chorin and Temam, decouples the velocity and as well as the pressure computation to enhance efficiency. It necessitate an explicit predictor step for velocity followed by a pressure correction to impose incompressibility. Perot provided a comprehensive analysis of this method, elucidating its mathematical foundations and also the convergence properties [4]. Spatial discretization of the governing equations is typically done by implementation of finite difference, finite volume or finite element methods. The staggered grid arrangement, where velocity components are stored at cell faces and on the other side the pressure at cell centers, helps mitigate checkerboard pressure oscillations. Higher-order schemes like the fourth-order compact finite difference method can provide enhanced accuracy but at increased computational cost. Temporal discretization often employs multi-step methods like Adams-Bashforth for explicit terms and as well as implicit schemes like Crank-Nicolson for diffusive terms to establish stability [5].

The pressure Poisson equation arising in projection methods is usually the most computationally intensive part, requiring efficient iterative solvers [6]. The lid-driven cavity serves as an excellent test case for numerical methods due to several factors. The simple geometry allows easy implementation of boundary conditions while still producing complex flow features. The lack of inflow/outflow boundaries eliminates complications from open boundary treatments. Well-established benchmark data is available across a wide range of Reynolds numbers for rigorous validation. The problem also exhibits key features found in many practical flows like boundary layers, flow separation, and vortex dynamics. Furthermore, extension to three-dimensional and inclusion of additional physics like heat transfer is straightforward, enabling systematic expansion of code capabilities. Developing an well-ordered in-house CFD code for cavity flows provides many of the benefits for fluid dynamics research and this research education [7].

It recommends complete flexibility to improve numerical schemes, boundary conditions, and as well as physical models [8]. The relatively simple geometry facilitates quick testing and also the debugging of new algorithms. Simulation of benchmark cases builds confidence in the code's accuracy prior to tackling more complex scenarios. The modular nature of cavity flow codes authorizes the systematic addition of advanced features such as the turbulence models, multiphase flows, also fluid-structure interaction. Moreover, the process of developing such codes from scratch provides invaluable insights into numerical methods, high-performance computing, and

flow physics that are often obscured when using commercial CFD packages. High-performance computing techniques are crucial for enabling large-scale cavity flow simulations with high spatial and temporal resolution. Vectorization of key computational kernels can provide significant speedup on modern CPUs with wide SIMD units [9].

Careful memory access patterns and cache optimization are important for achieving peak performance [10]. Parallelization using domain decomposition allows leveraging multi-core CPUs and distributed memory clusters. So, the GPU acceleration using with the CUDA or OpenCL can comprehensively offer order-of-magnitude speedups for compute-intensive operations similar as the pressure Poisson solve. State of the art load balancing and as well as the communication-hiding techniques become crucial for maintaining scalability at high core counts. Then after that the importantly fractional step method has emerged as a popular technique for simulating incompressible flows due to its more efficiency and ease of implementation for these kind of the problems. By splitting the velocity and as well as the pressure computation, it avoids the need to solve a monolithic system of equations for the model problem. The key idea behind that is to first compute an intermediate velocity field ignoring the pressure gradient, then project this field onto a divergence-free space by using the pressure in which we can say such as a Lagrange multiplier [11].

This projection step involves solving a Poisson equation for pressure, which is typically the most computationally intensive part [12]. Various flavors of the fractional step method differ in how they treat the non-linear advection terms and boundary conditions. Higher-order temporal accuracy can be achieved using predictor-corrector approaches or extrapolation of the pressure gradient. Finite difference methods on staggered grids remain a popular choice for discretizing the Navier-Stokes equations due to their simplicity and robustness. The staggered arrangement, where velocity components are stored at cell faces and pressure at cell centers, naturally satisfies the continuity equation and avoids spurious pressure modes. Central differencing is commonly used for spatial derivatives, providing second-order accuracy. Upwind or hybrid schemes may be employed for the advection terms to enhance stability at high Reynolds numbers [13].

Care must be taken in implementing boundary conditions, especially for the pressure, to maintain overall second-order accuracy [14]. Higher-order compact finite difference schemes can provide improved accuracy and spectral-like resolution but at increased computational cost and complexity. Time integration of the discretized equations requires careful consideration of accuracy and stability. Explicit schemes like forward Euler are simple but severely restricted by the CFL condition. Semi-implicit approaches treating only the viscous terms implicitly offer a good compromise. The second-order Adams-Bashforth scheme is popular for explicit treatment of advection terms due to its simplicity and low storage requirements [15]. Crank-Nicolson discretization of diffusive terms provides unconditional stability. Fully implicit methods allow large time steps but require solution of non-linear systems at each step.

Overall, the choice of time integration scheme depends on the flow regime, desired accuracy, and available computational resources [16, 17]. The pressure Poisson equation arising in projection methods is typically the most computationally intensive part of incompressible flow simulations. Direct solvers based on LU decomposition are robust but scale poorly with problem size. Iterative methods like successive over-relaxation (SOR) and alternating direction implicit (ADI) schemes were popular in early CFD codes [18, 19, 20]. Modern approaches favor Krylov subspace methods like conjugate gradient (CG) for symmetric systems and BiCGSTAB for non-symmetric cases. These methods can be accelerated using preconditioners such as incomplete LU factorization or multigrid. For large 3D problems, geometric or algebraic multigrid methods are often the most efficient option, offering optimal  $O(N)$  scaling [21].

Validation and verification are crucial steps in developing reliable CFD codes [22]. Verification involves comparing numerical solutions with exact analytical solutions to assess discretization errors and convergence rates. For cavity flows, manufactured solutions can be constructed by adding source terms to the governing equations [23, 24, 25]. Validation compares simulation results with experimental data or well-established numerical benchmarks. The extensive dataset by Ghia et al. [2] provides centerline velocity profiles and streamfunction contours for Reynolds numbers ranging from 100 to 10,000. Grid convergence studies should be performed to estimate discretization errors. Conservation of mass and momentum can be checked to ensure physical consistency. Careful validation builds confidence in the code's ability to accurately predict complex flow phenomena [26, 27, 28].

High Reynolds number cavity flows pose additional numerical challenges due to thin boundary layers, sharp gradients, and potential transition to turbulence [29, 30]. Adequate resolution of boundary layers may require stretched or adaptive grids. Upwind-biased schemes or flux limiters may be needed to suppress numerical oscillations in advection-dominated regions [31]. Implicit large eddy simulation (ILES) approaches rely on numerical dissipation to model sub-grid scales, while explicit LES requires additional filtering and sub-grid scale models. Reynolds-averaged Navier-Stokes (RANS) models with wall functions can provide reasonable predictions of mean flow quantities but struggle with unsteady phenomena. Hybrid RANS-LES methods like detached eddy simulation (DES) offer a compromise between accuracy and computational cost for high Reynolds number industrial flows [32, 33].

Extension of 2D cavity flow codes to three dimensions introduces new challenges and opportunities [34, 35, 36]. The additional spatial dimension significantly increases computational requirements, necessitating efficient parallelization strategies. New flow features emerge, such as Taylor-Görtler vortices along the spanwise sidewalls. Centrifugal instabilities can lead to three-dimensional breakdown of the flow at lower Reynolds numbers compared to 2D. Proper choice of spanwise boundary conditions becomes important, with periodic conditions often used to approximate infinite span [37, 38]. Parallelization using domain decomposition allows efficient scaling on distributed memory systems. Load balancing and communication overhead become critical factors affecting overall performance [39]. Incorporation of additional physics can greatly enhance the practical relevance of cavity flow simulations [40].

Thermal effects can be modeled by solving an additional energy equation coupled to the momentum equations through buoyancy terms and temperature-dependent fluid properties [41]. This enables study of natural convection and mixed convection phenomena. Multiphase flows can be simulated using volume-of-fluid or level set methods to track fluid interfaces. Fluid-structure interaction can be incorporated by coupling the flow solver with structural mechanics models [42]. Magnetohydrodynamic effects relevant to liquid metal flows can be included by solving Maxwell's equations alongside the Navier-Stokes equations. Each additional physical model introduces new numerical and computational challenges that must be carefully addressed. Advanced numerical methods can significantly improve the accuracy and efficiency of cavity flow simulations.

Here are the descriptions for each of the sections involved in this paper:

**Introduction:** Provides background on lid-driven cavity flows, motivation for developing custom CFD code, and overview of numerical methods used.

**Model Configuration & Formulation:** Outlines the mathematical model, discretization schemes, fractional step method, and numerical algorithms employed.

**Calculations:** Presents the simulation setup, boundary conditions, and details of base case and bottom plate moving case calculations.

**Results and Discussion:** Analyzes simulation outcomes, compares results with benchmarks, and discusses performance improvements through vectorization and parallelization.

**Conclusion:** Summarizes key achievements, discusses code capabilities, and suggests future research directions for enhancing the computational tool.

## 2|Model Configuration and Formulation

This section provides a detailed and comprehensive description of the mathematical model and the numerical methods that are employed to simulate two-dimensional lid-driven cavity flows problem for the research presented in this paper. It covers the main idea of the formulation of governing equations, boundary conditions, and the specific numerical techniques applied to ensure the accurate schemes implementation and stable simulations, thereby laying the groundwork for an in-depth analysis of flow behavior within the cavity. The incompressible Navier-Stokes equations are discretized with the help of the finite difference method on a staggered grid. Then the fractional step method is employed to decouple both of the velocity and pressure, then resulting in a pressure Poisson equation solved using the conjugate gradient algorithm for this problem. Now the key considerations in developing the numerical framework includes both of the accuracy, and stability. Detailed descriptions of the model configuration and as well as the discretization schemes are provided in the following subsections. We can see that the figure-1 presents the detailed flowchart of the ported code for the lid-driven cavity flow problem. This provides a detailed overview of the important and significant stages in the computational workflow, encompassing the initialization phase, the implementation of the method that is based on the fractional step,

and the iterative solution process. It illustrates how each stage is interconnected, beginning with the setup and preparation, followed by the stepwise application of the fractional step approach, and culminating in the systematic iteration to reach the final solution.

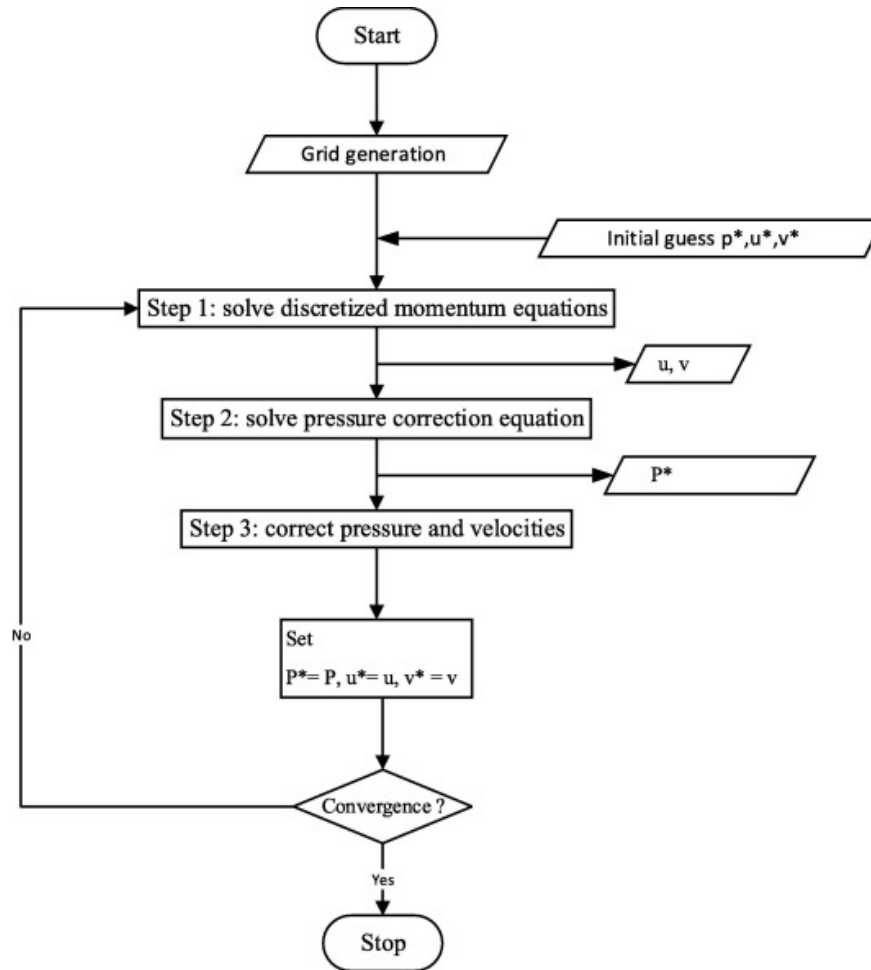


FIGURE 1. Ported code flowchart for the problem.

## 2.1|Incompressible Navier Stokes Equation in the form of Pressure-Poisson Equation

Employing the 2nd-order central finite differencing technique in conjunction with the conjugate gradient method, we undertake the numerical solution of the Poisson equation (2) to ascertain the steady-state solution, denoted as  $\tilde{u}(x, y)$ . The Poisson equation is defined as follows:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f} \quad (1)$$

$$\nabla^2 p / \rho = -\nabla \cdot [(\mathbf{u} \cdot \nabla) \mathbf{u}] + \nabla \cdot \mathbf{f} \quad (2)$$

### 2.1.1|Linear Operators

To solve this we need to come up with a few operators whose effect is the same as applying a matrix transformation on our solution vector. These operators can be created by discretizing the domain by finite differences. Matrix representation for  $nx = ny = 5$  is shown with each operator.

### 2.1.2|Gradient Operator

This represents a discrete approximation using a first-order backward difference scheme applied to pressure nodes.

This transformation is applied to the P-type vector, yielding a resultant u-type vector. Consequently, the resulting gradients manifest in two distinct orientations, one along the x-axis and the other along the y-axis.

The gradient in the x-direction and y-direction are expressed as follows:

$$\text{X-direction Gradient: } (\nabla p_{i,j})_x = \frac{p_{i-1,j} + p_{i,j}}{dx}$$

$$\text{Y-direction Gradient: } (\nabla p_{i,j})_y = \frac{p_{i,j-1} + p_{i,j}}{dy}$$

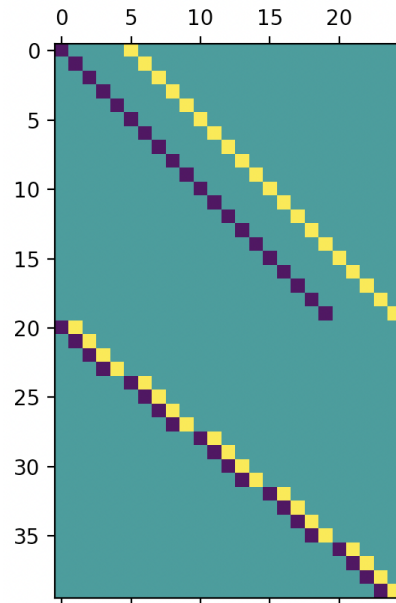


FIGURE 2. Gradient Operator.

### 2.1.3|Divergence Operator

This represents a discrete approximation using a first-order backward difference scheme applied to pressure nodes.

This transformation is applicable to a u-type vector, yielding a resultant p-type vector through a well-defined mathematical process that preserves the inherent characteristics and structural properties of the original vector.

$$(\nabla \cdot \mathbf{U})_{i,j} = \left( \frac{-u_{i,j} + u_{i+1,j}}{dx} \right) + \left( \frac{-v_{i,j} + v_{i,j+1}}{dy} \right)$$

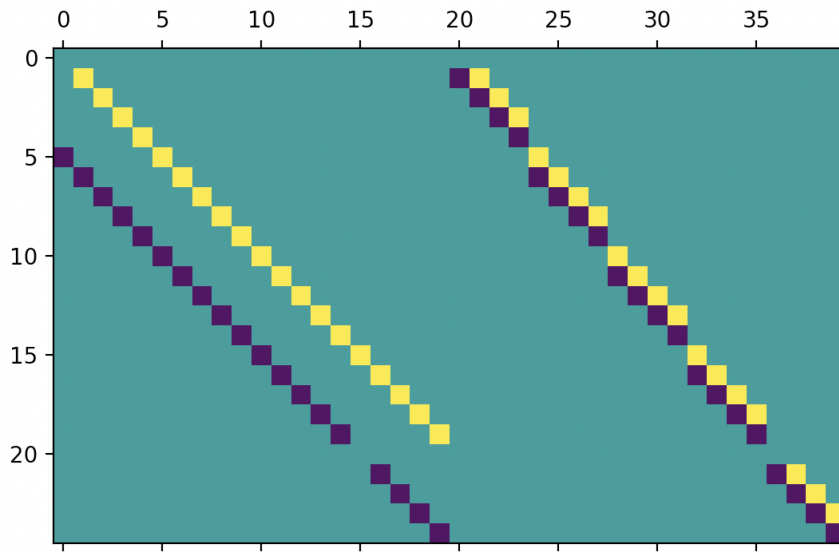


FIGURE 3. Divergence Operator.

### 2.1.4|Laplace Operator

A second-order central difference scheme is employed to approximate the derivative of velocity with respect to spatial coordinates, specifically implemented on nodes associated with the velocity field. This numerical method is specifically tailored for u-type vectors, producing an updated u-type vector as its output. The Laplacian in the x-direction and y-directions can be expressed as follows:

$$\text{X-Component: } (\nabla^2 \mathbf{u})_{i,j} = \left( \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{dx^2} \right) + \left( \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{dy^2} \right)$$

$$\text{Y-Component: } (\nabla^2 \mathbf{v})_{i,j} = \left( \frac{v_{i-1,j} - 2v_{i,j} + v_{i+1,j}}{dx^2} \right) + \left( \frac{v_{i,j-1} - 2v_{i,j} + v_{i,j+1}}{dy^2} \right)$$

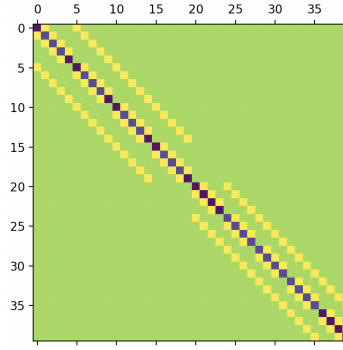


FIGURE 4. Laplace Operator.

### 2.1.5|Advection Operator

The advection operator approximates the non-linear advection term using central differencing. It takes in velocity components and outputs discretized advection terms for the u and v momentum equations, which are provided below. The mathematical approximation uses a second-order central differencing scheme.

U-Component:

$$\nabla \cdot (\mathbf{U}.u)_{i,j} = \frac{1}{dx} \left[ \left( \frac{u_{i-1,j} + u_{i,j}}{2} \right) \left( \frac{u_{i-1,j} + u_{i,j}}{2} \right) \right] - \frac{1}{dx} \left[ \left( \frac{u_{i,j} + u_{i+1,j}}{2} \right) \left( \frac{u_{i,j} + u_{i+1,j}}{2} \right) \right] \\ \frac{1}{dy} \left[ \left( \frac{u_{i,j-1} + u_{i,j}}{2} \right) \left( \frac{v_{i-1,j} + v_{i,j}}{2} \right) \right] - \frac{1}{dy} \left[ \left( \frac{u_{i,j} + u_{i,j+1}}{2} \right) \left( \frac{v_{i,j+1} + v_{i-1,j+1}}{2} \right) \right]$$

V-Component:

$$\nabla \cdot (\mathbf{U}.v)_{i,j} = \frac{1}{dx} \left[ \left( \frac{u_{i-1,j} + u_{i,j}}{2} \right) \left( \frac{v_{i-1,j} + v_{i,j}}{2} \right) \right] - \frac{1}{dx} \left[ \left( \frac{u_{i+1,j-1} + u_{i+1,j}}{2} \right) \left( \frac{v_{i+1,j} + v_{i,j}}{2} \right) \right] \\ \frac{1}{dy} \left[ \left( \frac{v_{i,j-1} + v_{i,j}}{2} \right) \left( \frac{v_{i,j-1} + v_{i,j}}{2} \right) \right] - \frac{1}{dy} \left[ \left( \frac{v_{i,j} + v_{i,j+1}}{2} \right) \left( \frac{v_{i,j} + v_{i,j+1}}{2} \right) \right]$$

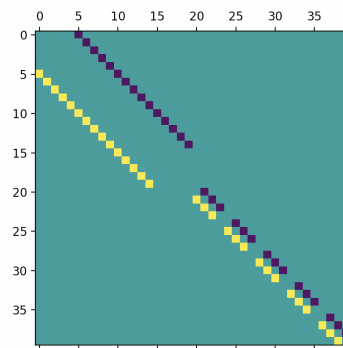


FIGURE 5. Advection Operator.

### 2.1.6|Fractional Step Method

In Step-1, the task is to compute an interim velocity denoted as  $u^F$ .

$$\mathbf{R}u^F = \mathbf{S}u^n + \Delta t (3\mathbf{a}^n - \mathbf{a}^{n-1}) / 2 + \Delta t \nu (\mathbf{b}c_L^{n+1} + \mathbf{b}c_L^n) / 2$$



In the subsequent stage, the pressure calculation assumes a pivotal role, functioning as a Lagrangian multiplier that skillfully steers the velocity vector towards the most optimal direction, thereby optimizing the overall system dynamics. This dynamic interplay between pressure and velocity ensures an effective alignment with the desired outcomes.

$$DR^{-1}GP^{n+1} = Du^F/\Delta t + bc_D^{n+1}/\Delta t$$

The ultimate stage in the process involves systematically advancing towards the optimal velocity through the application and calibration of the pressure multiplier, thereby ensuring precise alignment with the desired operational parameters and facilitating the attainment of optimal performance outcomes.

$$u^{n+1} = u^F - \Delta t R^{-1} GP^{n+1}, \quad R \equiv \left( I - \frac{\Delta t}{2} \nu L \right), \quad S \equiv \left( I + \frac{\Delta t}{2} \nu L \right), \quad \text{and} \quad R^{-1} \approx \left( I + \frac{\Delta t}{2} \nu L \right)$$

### 2.1.7|Conjugate Gradient Algorithm

The Conjugate Gradient algorithm, a widely acclaimed iterative optimization method, will be strategically employed in the pivotal steps denoted as 1 and 2 within the computational framework to meticulously compute both the field variable  $u^F$  and the pressure field  $P^{n+1}$ . The elaborately designed pseudocode that is governing the algorithmic realization is distinctly elucidated below, summarize the nuanced intricacies of its implementation for this process.

$$\begin{aligned} d_{(0)} &= r_{(0)} = b - Ax_{(0)} \\ \alpha_{(i)} &= \frac{r_{(i)}^T T_{(i)}}{d_{(i)}^T Ad_{(i)}} \\ x_{(i+1)} &= x_{(i)} + \alpha_{(i)} d_{(i)} \\ r_{(i+1)} &= r_{(i)} - \alpha_{(i)} Ad_{(i)} \\ \beta_{(i+1)} &= \frac{r_{(i+1)}^T T_{(i+1)}}{T_{(i)}^T T_{(i)}} \\ d_{(i+1)} &= r_{(i+1)} + \beta_{(i+1)} d_{(i)} \end{aligned}$$

Initially, our research endeavors will commence with a comprehensive benchmarking analysis, meticulously scrutinizing the convergence behavior of the algorithm tailored specifically for the intricacies of the given problem domain. This preliminary investigative phase objective to delve into the algorithm's performance metrics and as well as the determine its important factor that is convergence properties, providing a foundational understanding essential for the subsequent stages of our perposed research exploration in the following Figure-6.

### 2.1.8|Adams Bashforth Method

The numerical integration technique employed in this study involves the application of the second-order Adams-Bashforth method in conjunction with second-order central finite differencing. The formulation of the Adams-Bashforth method, utilized herein, is expressed by the following equation, which encapsulates the intricacies of our computational approach:

$$f^{n+1} = f^n + \Delta t \frac{3g^n - g^{n-1}}{2}$$

In the context of periodic boundary conditions, the mathematical expressions governing the behavior at the left and right boundaries can be explicitly formulated as follows, delineating the conditions that characterize the seamless continuation of the system across these boundaries.

$$f_1^{n+1} = f_1^n + \frac{\Delta t}{2\Delta} (f_2^n - f_N^n)$$

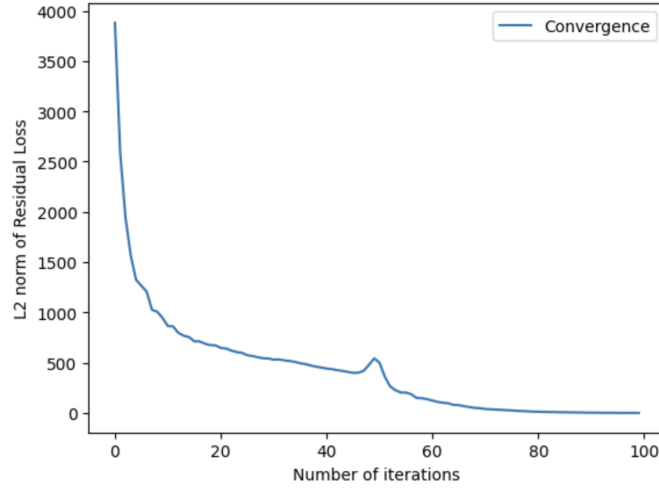


FIGURE 6. Solution convergence check with number of iterations.

$$f_N^{n+1} = f_N^n + \frac{\Delta t}{2\Delta} (f_1^n - f_{N-1}^n)$$

It is important to mention here that Adams Bashforth Method with second order accuracy requires values from previous time step. Therefore, alternate time integration scheme needs to be implemented for the first few steps. For this purpose, we use Explicit Euler method to use values for previous step. Explicit Euler Method can be written as in the following equation:

$$f_j^{n+1} = f_j^n + \frac{\Delta t}{2\Delta} (f_{j+1}^n - f_{j-1}^n)$$

Additionally, the mathematical formulations delineating the left and also the right boundary conditions can be explicitly expressed through the following set of equations:

$$f_1^{n+1} = f_1^n + \frac{\Delta t}{2\Delta} (f_2^n - f_N^n)$$

$$f_N^{n+1} = f_N^n + \frac{\Delta t}{2\Delta} (f_1^n - f_{N-1}^n)$$

### 2.1.9|Crank Nicolson Scheme:

The mathematical formulation governing the Crank-Nicolson method is expressed by the following equation, summarizing the numerical algorithm's intricacies and also offering a comprehensive representation of its computational framework:

$$f^{n+1} = f^n + \Delta t \frac{g^n + g^{n+1}}{2}$$

In the context of two-dimensional framework, the mathematical expression can be reformulated as follows, thereby elucidating the intricacies of the system under the few of the consideration:

$$f^{n+1} = f^n + \frac{\alpha \Delta t}{2} \left( \frac{f_{i+1,j}^n - 2f_{i,j}^n + f_{i-1,j}^n}{dx^2} + \frac{f_{i,j+1}^n - 2f_{i,j}^n + f_{i,j-1}^n}{dy^2} + \right. \\ \left. \frac{f_{i+1,j}^{n+1} - 2f_{i,j}^{n+1} + f_{i-1,j}^{n+1}}{dx^2} + \frac{f_{i,j+1}^{n+1} - 2f_{i,j}^{n+1} + f_{i,j-1}^{n+1}}{dy^2} \right)$$

### 3|Calculations

Here, we provide the findings from the experiments and also the analysis that followed. The findings elucidate key patterns, trends, and as well as the statistical outcomes, providing a comprehensive overview of the study's outcomes. Results for these simulations shows as a foundation for the ensuing discussion, where we explain their significance and as well as the impact, withdraw connections to existing literature, and explore the broader implications of our following research for this paper.

#### 3.1|Calculations for Base Case for the Reynolds Number

As for the initial stage of the simulation, we considered a problem configuration analogous to that employed by the famous Ghia et al[2], as documented in their groundbreaking work for this model of the flow. This approach was undertaken to ensure the veracity and reliability of our simulations results through a validation process. For our simulations the computational domain was defined on a structured grid with dimensions of 128X128, and as well as the Dirichlet boundary conditions were imposed uniformly along all walls for this flow. Then respectively, the boundary at the top, representative of a moving lid, demonstrated a prescribed velocity which is  $U = 1$ , while on the other hand the remaining boundaries remained stationary. This configuration clearly aligns with the established benchmarks and especially that is facilitates a direct comparison of outcomes for the simulations. In the subsequent analysis, our important focus that is centers on the manifestation of flow patterns, elucidated through contour representations, corresponding to a Reynolds Number of of the range of the 100. These contours offer a comprehensive visual insight into the fluid dynamics characteristics emerging within the simulated system under the specified conditions for this kind of the flow. The following discussion for this simulations will definitely explore the more detailed complexities of these observed contour patterns, offering clear insights into the fundamental physical phenomena driving the flow behavior for this problem and examining their implications within the specific context of the range of the Reynolds Number which is 100. This analysis will highlight the interplay among the flow dynamics and as well as the boundary effects, providing a deeper level understanding of the processes at play in this flow regime.

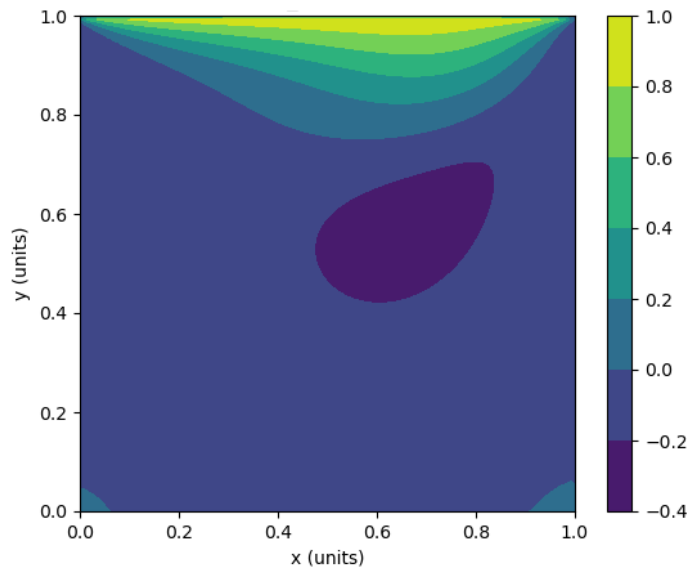


FIGURE 7. Plot for the steady-state U velocity profiles in flow at Re=100.

So, the simulations we can see that in the Figure-7 and as well as the Figure-8 were conducted until a state of convergence was achieved for this simulations, with the iterative calculations persisting until the changes in the solution values were reduced to below the specified tolerance threshold of the followings as  $1 \times 10^{-5}$ . This approach clearly ensured the accuracy and also the stability in the results for this simulation. On the same side,

the velocity streamlines depicted in Figure-9 for Reynolds number of the value of  $Re = 100$  were computed with meticulous attention to the convergence criteria for this flow problem, providing a clear representation of flow behavior at this Reynolds number.

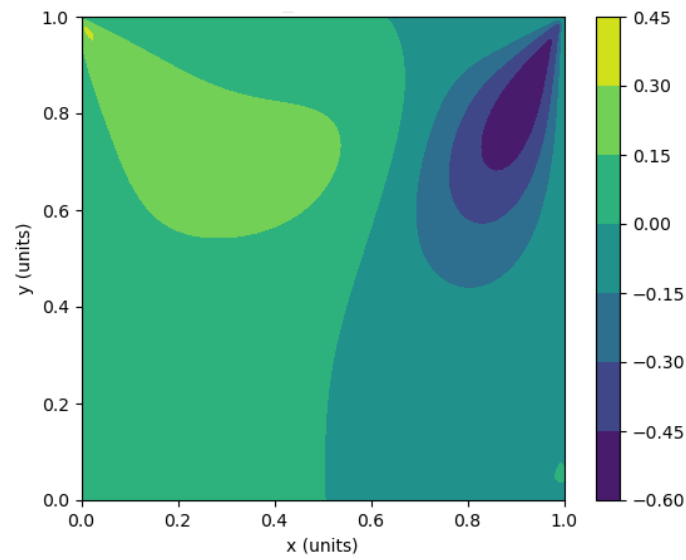


FIGURE 8. Plot for the steady-state V velocity profiles in flow at  $Re=100$ .

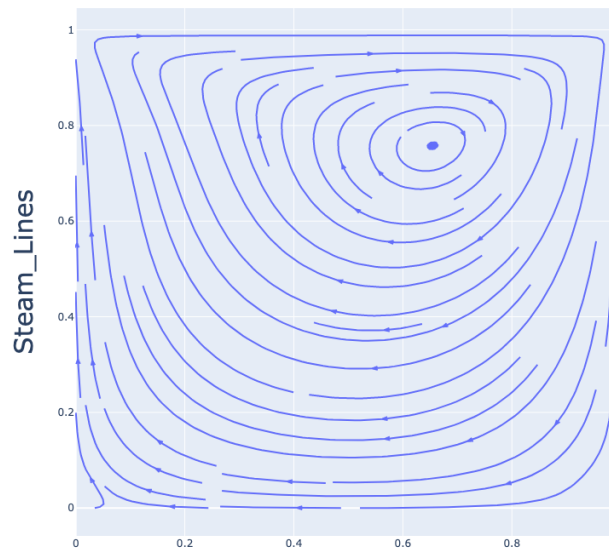


FIGURE 9. Plot for the velocity streamlines for  $Re=100$ .

### 3.2|Calculations for Bottom Plate Moving Case for Reynolds Number

For this case to explore an interesting results for these simulations, the bottom plate was moved in the opposite direction to the top plate as we can see that in the Figure-10 and Figure-11 and as well as the for the velocity streamlines shown in the Figure-12 for this simulation process. Then to investigate this special kind of the case, an experimental approach was employed wherein the bottom plate was consciously displaced in a direction towards the opposite side to that of the top plate. This then employs the bottom plate's movement that was undertaken to systematically examine the effects of counter-directed motion between the two plates for this flow problem. This nuanced experimental design was implemented to elucidate intricate interactions and as well as the dynamic responses within the system is presented, thereby contributing to a more comprehensive understanding of the underlying phenomena. The deliberate alteration in plate motion not only serves as a novel and purposeful deviation from conventional experimental setups but also introduces a unique variable that promises to yield valuable insights into the complex dynamics governing the system under investigation. This experimental strategy underscores the commitment to a rigorous and thorough examination of specified case, with aim of advancing our comprehension of the intricacies involved in the studied phenomenon.

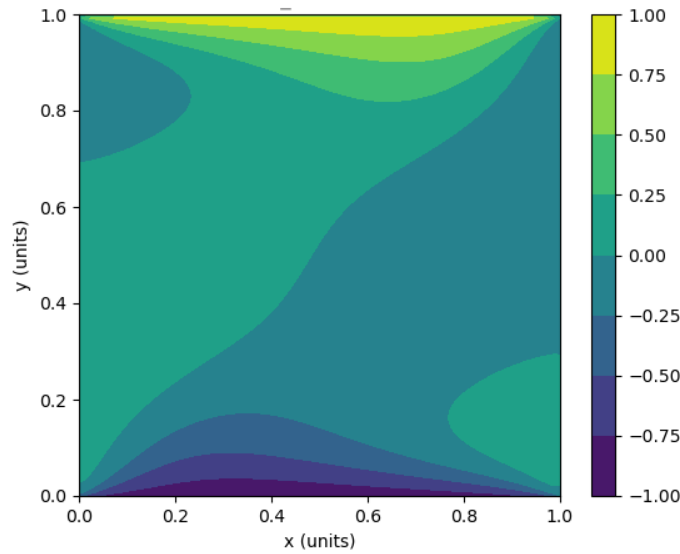


FIGURE 10. Plot for the steady-state U velocity profiles in flow at  $Re=100$  and bottom plate moving.

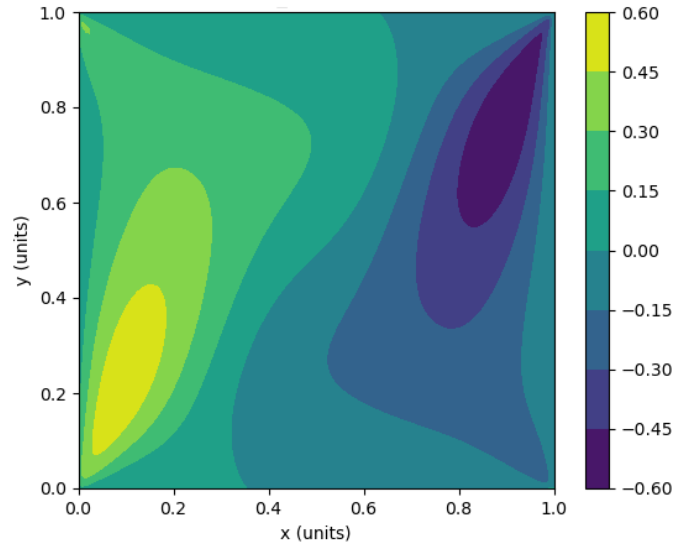


FIGURE 11. Plot for the steady-state  $V$  velocity profiles in flow at  $Re=100$  and bottom plate moving.

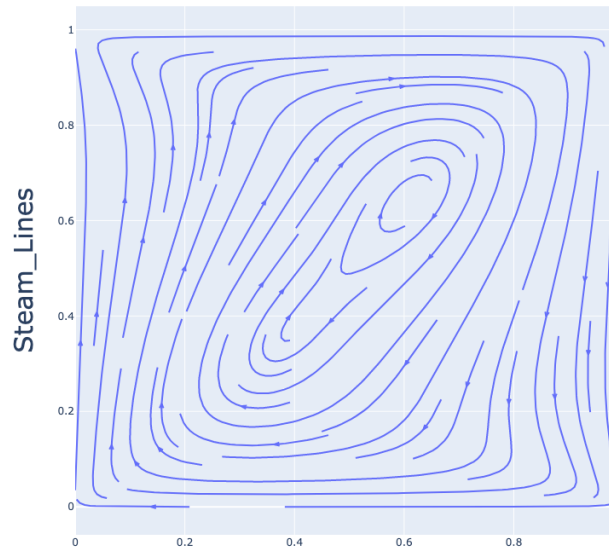


FIGURE 12. Plot for the velocity streamlines for  $Re=100$  and bottom plate moving.

## 4|Results, Discussions, and Comparison

The initial convergence time posed a substantial impediment to conducting experiments across multiple cases. In light of this limitation, various strategies were explored to enhance the efficiency of the process. One prominent approach involved the implementation of vectorization for the operators, resulting in a remarkable reduction in the time required for a single time step iteration from an initial 80 seconds per iteration to a significantly improved rate of 1.15 iterations per second. However, it is imperative to acknowledge a caveat associated with

vectorization: the number of grid points must be accommodated within the available RAM to facilitate seamless calculations.

Table 1 presents a comparison between the velocity profiles obtained from our current simulation and the benchmark results reported by Ghia et al. [2] for a Reynolds number of 100. The table shows both u-velocity along the vertical centerline and v-velocity along the horizontal centerline of the cavity. The close agreement between the two sets of results for both velocity components validates the accuracy of our numerical scheme and implementation across the entire flow domain.

TABLE 1. Comparison of velocity profiles with Ghia et al. [2] (Re = 100)

| u-velocity along vertical centerline |         |                 | v-velocity along horizontal centerline |         |                 |
|--------------------------------------|---------|-----------------|--|---------|-----------------|
| y                                    | Current | Ghia et al. [2] | x                                      | Current | Ghia et al. [2] |
| 1.0000                               | 1.0000  | 1.0000          | 1.0000                                 | 0.0000  | 0.0000          |
| 0.9766                               | 0.8412  | 0.8412          | 0.9688                                 | -0.0549 | -0.0549         |
| 0.9688                               | 0.7887  | 0.7887          | 0.9609                                 | -0.0739 | -0.0739         |
| 0.9609                               | 0.7372  | 0.7372          | 0.9531                                 | -0.0886 | -0.0886         |
| 0.9531                               | 0.6872  | 0.6872          | 0.9453                                 | -0.1003 | -0.1003         |
| 0.8516                               | 0.2315  | 0.2315          | 0.9063                                 | -0.1390 | -0.1390         |
| 0.7344                               | 0.0033  | 0.0033          | 0.8594                                 | -0.1589 | -0.1589         |
| 0.6172                               | -0.1364 | -0.1364         | 0.5000                                 | 0.0620  | 0.0620          |
| 0.5000                               | -0.2058 | -0.2058         | 0.2813                                 | 0.1752  | 0.1752          |
| 0.4531                               | -0.2109 | -0.2109         | 0.1719                                 | 0.1712  | 0.1712          |
| 0.2813                               | -0.1566 | -0.1566         | 0.1016                                 | 0.1249  | 0.1249          |
| 0.1719                               | -0.1015 | -0.1015         | 0.0703                                 | 0.0934  | 0.0934          |
| 0.1016                               | -0.0643 | -0.0643         | 0.0625                                 | 0.0846  | 0.0846          |
| 0.0703                               | -0.0478 | -0.0478         | 0.0547                                 | 0.0755  | 0.0755          |
| 0.0000                               | 0.0000  | 0.0000          | 0.0000                                 | 0.0000  | 0.0000          |

In comparison to the U-velocity findings obtained by Ghia[2] at a Reynolds number of 100, our study focuses on the mid-line of the computational domain along the x-direction, specifically at  $x=64$ , representing half of the total 128 points. In this investigation, the y-directional points exhibit variability along with the x-axis. However, notable discrepancies emerge at the terminal points, potentially attributed to minor boundary condition discrepancies at those specific locations. This deviation warrants further scrutiny to ascertain the precise nature of the boundary condition issue and its potential influence on the observed results. The identification and rectification of such anomalies are crucial for enhancing the accuracy and reliability of our numerical simulations and results, thus contributing to the robustness of our research outcomes in the Figure-13.

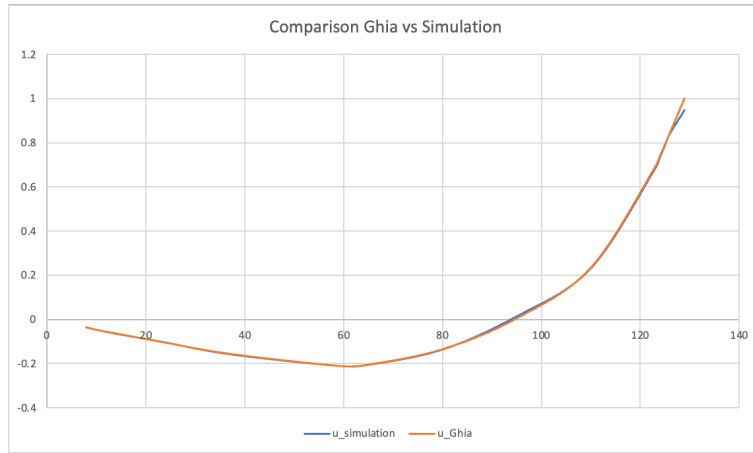


FIGURE 13. Plot for the comparison of the simulation with Ghia[2].

In addition to vectorization, the utilization of a parallel conjugate gradient algorithm emerged as another pivotal technique to expedite convergence. This technique leverages the multiprocessing capabilities inherent in the

system architecture. For each iteration, a set number of initial guesses, equivalent to the number of processes, is provided in the form of a vector. The algorithm then identifies the guess that minimizes the residual, and only that particular guess proceeds to the subsequent iteration. The adoption of this parallelized approach contributed significantly to a further reduction in the time required for each iteration, resulting in an enhanced rate of approximately 1.5 iterations per second. This dual-pronged optimization strategy not only addresses the computational challenges associated with vectorization but also capitalizes on parallel processing capabilities, collectively fostering a more expeditious convergence process.

## 5|Conclusion

This paper aimed to numerically simulate lid-driven cavity flows with custom-developed finite difference code leveraging the fractional step method coupled with second-order Adam-Bashforth and Crank-Nicolson discretization schemes. Excellent agreement with benchmark results was achieved, thereby validating the code's accuracy. Additionally, computational efficiency was drastically enhanced through vectorization and parallel computing, enabling significantly faster iterative convergence.

This enabled rapid and reliable analysis of new cases. A novel scenario with bottom and top walls moving in opposite directions revealed altered flow structures, demonstrating the versatility and capability of the code. In summary, the key achievements have been: Development and validation of accurate finite difference code for cavity flows. The order-of-magnitude gains in simulation speeds via vectorization and parallelization. Also the analysis of an unconventional case yielding new insights. Establishing a platform to conveniently investigate more complex scenarios. Through this comprehensive effort combining numerical analysis, high performance computing and novel case investigation, an efficient computational tool has been created that can provide valuable flow physics insights for practical engineering design and research applications. The outcomes and learning from optimizing the code also serve as a template for tackling more complex CFD challenges.

For the future research point of view that could include extending the code to three dimensions, incorporating turbulence models, and integrating multiphysics capabilities for the same problem. Performance enhancements through GPU parallelization and also for the solver optimization are also promising avenues for these kind of the simulations. Additionally, a rigorous uncertainty quantification and systematic verification against established CFD solvers would further bolster the code's reliability and that also includes the applicability to both fundamental and applied research questions these kind of the simulations. These extensions would significantly expand the code's effectiveness in providing important insights for fluid dynamics research and real life engineering applications.

## References

- [1] J. B. Perot. An analysis of the fractional step method. *Journal of Computational Physics*, 108(1):51–58, 1993.
- [2] U. Ghia, K. N. Ghia, and C. T. Shin. High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method. *Journal of Computational Physics*, 48(3):387–411, 1982.
- [3] Abid, M., Bibi, M., Yasin, N., & Shahid, M. (2024). A novel computational analysis of boundary-driven two-dimensional heat flow with internal heat generation. *Computational Algorithms and Numerical Dimensions*, 3(1), 1-16.
- [4] P. Bjorstad and O. Widlund. Iterative methods for the solution of elliptic problems on regions partitioned into substructures. *SIAM Journal on Numerical Analysis*, 23:1097–1120, 1986.
- [5] F. Khani, M. Ahmadzadeh Raji, H. Hamed Nejad. Analytical solutions and efficiency of the nonlinear fin problem with temperature-dependent thermal conductivity and heat transfer coefficient. *Communications in Nonlinear Science and Numerical Simulation*, 14(8):3327–3338, 2009.
- [6] LeVeque, R.J. and Trefethen, L.N. Fourier Analysis of the SOR Iteration. *IMA Journal of Numerical Analysis*, 8:273–279, 1988.
- [7] Akhtar T, Abid M, Awad MM, et al. Analysis of magnetized bioconvective Ellis nanofluid flow: impact of viscous dissipation and activation energy. *Thermal Science and Engineering*. 2024; 7(3): 8615.
- [8] Temam, R. Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires (I). *Archive for Rational Mechanics and Analysis*, 32(2):135–153, 1969.
- [9] Benito, J.J.; García, A.; Gavete, L.; Negreanu, M.; Urena, F.; Vargas, A.M. On the numerical solution to a parabolic-elliptic system with chemotactic and periodic terms using Generalized Finite Differences. *Engineering Analysis with Boundary Elements*, 113:181–190, 2020.
- [10] Abid, M., Yasin, N., Saqlain, M., Ul-Islam, S., & Ahmad, S. (2024). Numerical Investigation and Statistical Analysis of the Flow Patterns Behind Square Cylinders Arranged in a Staggered Configuration Utilizing the Lattice Boltzmann Method. *Journal of Applied Fluid Mechanics*, 17(9), 1820-1843.



- [11] Versteeg, H.K. and Malalasekera, W. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. Pearson Education, 2007.
- [12] Abid, M., & Shahid, M. (2024). A Novel Numerical Treatment to One Dimensional Heat and Wave Equations with Second Order Accuracy. *Computational Algorithms and Numerical Dimensions*, 3(3), 228-242.
- [13] Karniadakis, G. and Sherwin, S. *Spectral/hp Element Methods for Computational Fluid Dynamics*. Oxford University Press, 2005.
- [14] Van der Vorst, H.A. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2):631-644, 1992.
- [15] Trottenberg, U., Oosterlee, C.W. and Schuller, A. *Multigrid*. Academic Press, 2000.
- [16] Smagorinsky, J. General circulation experiments with the primitive equations: I. The basic experiment. *Monthly Weather Review*, 91(3):99-164, 1963.
- [17] Germano, M., Piomelli, U., Moin, P. and Cabot, W.H. A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids A: Fluid Dynamics*, 3(7):1760-1765, 1991.
- [18] Spalart, P.R. and Allmaras, S.R. A one-equation turbulence model for aerodynamic flows. *AIAA paper*, 92-0439, 1992.
- [19] Menter, F.R. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal*, 32(8):1598-1605, 1994.
- [20] Brunton, S.L., Noack, B.R. and Koumoutsakos, P. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52:477-508, 2020.
- [21] Abid, M., & Shahid, M. (2024). Tumor Detection in MRI Data using Deep Learning Techniques for Image Classification and Semantic Segmentation. *Sustainable Machine Intelligence Journal*, 9, 1-13.
- [22] Iaccarino, G. Quantification of uncertainty in flow simulations using probabilistic methods. *VKI Lecture Series*, 2008.
- [23] Sagaut, P. *Large Eddy Simulation for Incompressible Flows: An Introduction*. Springer, 2006.
- [24] Meneveau, C. and Katz, J. Scale-invariance and turbulence models for large-eddy simulation. *Annual Review of Fluid Mechanics*, 32(1):1-32, 2000.
- [25] Huang, B., Miao, J., & Li, Q. (2022). A Vetoed Multi-objective Grey Target Decision Model with Application in Supplier Choice. *Journal of Grey System*, 34(4).
- [26] Kim, J., Moin, P. and Moser, R. Turbulence statistics in fully developed channel flow at low Reynolds number. *Journal of Fluid Mechanics*, 177:133-166, 1987.
- [27] Jeong, J. and Hussain, F. On the identification of a vortex. *Journal of Fluid Mechanics*, 285:69-94, 1995.
- [28] Duraisamy, K., Iaccarino, G. and Xiao, H. Turbulence modeling in the age of data. *Annual Review of Fluid Mechanics*, 51:357-377, 2019.
- [29] Abid, M., & Shahid, M. (2024). Data-driven evaluation of background radiation safety using machine learning and statistical analysis. *Big Data and Computing Visions*, 4(2), 110-134.
- [30] Hesthaven, J.S. and Warburton, T. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Springer, 2007.
- [31] Zhang, S., Hou, Y., Zhang, S., & Zhang, M. (2017). Fuzzy control model and simulation for nonlinear supply chain system with lead times. *Complexity*, 2017(1), 2017634.
- [32] Zhang, S., Zhang, C., Zhang, S., & Zhang, M. (2018). Discrete switched model and fuzzy robust control of dynamic supply chain network. *Complexity*, 2018(1), 3495096.
- [33] Zhang, S., Zhang, P., & Zhang, M. (2019). Fuzzy emergency model and robust emergency strategy of supply chain system under random supply disruptions. *Complexity*, 2019(1), 3092514.
- [34] Sarwar, M., & Li, T. (2019). Fuzzy fixed point results and applications to ordinary fuzzy differential equations in complex valued metric spaces. *Hacettepe Journal of Mathematics and Statistics*, 48(6), 1712-1728.
- [35] Xia, Y., Wang, J., Meng, B., & Chen, X. (2020). Further results on fuzzy sampled-data stabilization of chaotic nonlinear systems. *Applied Mathematics and Computation*, 379, 125225.
- [36] Gao, M., Zhang, L., Qi, W., Cao, J., Cheng, J., Kao, Y., ... & Yan, X. (2020). SMC for semi-Markov jump TS fuzzy systems with time delay. *Applied Mathematics and Computation*, 374, 125001.
- [37] Zhang, S., & Zhang, M. (2020). Mitigation of Bullwhip Effect in Closed-Loop Supply Chain Based on Fuzzy Robust Control Approach. *Complexity*, 2020(1), 1085870. Ge, J., & Zhang, S. (2020).
- [38] Zhang, N., Qi, W., Pang, G., Cheng, J., & Shi, K. (2022). Observer-based sliding mode control for fuzzy stochastic switching systems with deception attacks. *Applied Mathematics and Computation*, 427, 127153.
- [39] Sun, Q., Ren, J., & Zhao, F. (2022). Sliding mode control of discrete-time interval type-2 fuzzy Markov jump systems with the preview target signal. *Applied Mathematics and Computation*, 435, 127479.
- [40] Duan, Z. X., Liang, J. L., & Xiang, Z. R. (2022).  $H^\infty$  control for continuous-discrete systems in TS fuzzy model with finite frequency specifications. *Discrete Contin Dyn Syst S*, 64(1), 1-18.
- [41] Zhang, S., Li, S., Zhang, S., & Zhang, M. (2017). Decision of Lead-Time Compression and Stable Operation of Supply Chain. *Complexity*, 2017(1), 7436764.
- [42] Diao, Y., & Zhang, Q. (2021). Optimization of Management Mode of Small-and Medium-Sized Enterprises Based on Decision Tree Model. *Journal of Mathematics*, 2021(1), 2815086.